Rezoluce: další formální systém

- vhodná pro strojové dokazování (Prolog)
- metoda založená na *vyvracení*: dokazuje se nesplnitelnost formulí
- pracujeme s formulemi v nkf (též *klauzulárním tvaru*), ale používáme jinou notaci:
  - *klauzule* je konečná množina literálů (chápaná jako jejich disjunkce); je pravdivá, pokud alespoň jeden prvek je pravdivý. Prázdná klauzule $\square$ je vždy nepravdivá – neobsahuje žádný pravdivý prvek.
    Příklad klauzule: $\{p, \neg q, r\}$ (původně značeno $p \vee \neg q \vee r$)
  - *formule* je (ne nutně konečná) množina klauzulí (chápaná jako jejich konjunkce, tedy nkf); je pravdivá, je-li každý prvek pravdivý. Prázdná formule $\emptyset$ je vždy pravdivá – neobsahuje žádný nepravdivý prvek.
    Příklad formule: $\{\{\neg q\}, \{\neg p, q\}, \{p, q, r\}\}$ (původně značeno $\neg q \wedge (\neg p \vee q) \wedge (p \vee q \vee r)$)

Rezoluce – důkazy

- *rezoluční pravidlo*: nechť $C_1 = \{p\} \sqcup C_1', C_2 = \{\neg p\} \sqcup C_2'$ jsou klauzule, jejich *rezolventou* je $C = C_1' \cup C_2'$ (rezolvovali jsme na literálu p)
- rezoluční pravidlo zachovává splnitelnost (lib. valuace splňující C_1 a C_2 splňuje i C; klauzule C_1, C_2 označujeme jako *rodiče*, C jako *potomka*)
- *rezoluční důkaz* klauzule C z formule S je konečná posloupnost klauzulí $C_1, C_2, \ldots, C_n$, kde $C_n = C$ a každé C_i je buď prvkem S, nebo rezolventou klauzulí C_j, C_k pro $j, k < i$. Existuje-li tento důkaz, je C *rezolučně dokazatelná* z S (píšeme $S \vdash_R C$). Odvození $\square$ z S je *vyvrácením* S.
- příklad: dokažte $C = \{\neg q\}$ z $S = \{\{p, r\}, \{\neg q, \neg r\}, \{\neg p\}\}$
  $C_1 = \{p, r\}$ (prvek S), $C_2 = \{\neg q, \neg r\}$ (prvek S),
  $C_3 = \{p, \neg q\}$ (rezolventa C_1, C_2), $C_4 = \{\neg p\}$ (prvek S),
  $C = C_5 = \{\neg q\}$ (rezolventa C_3, C_4)

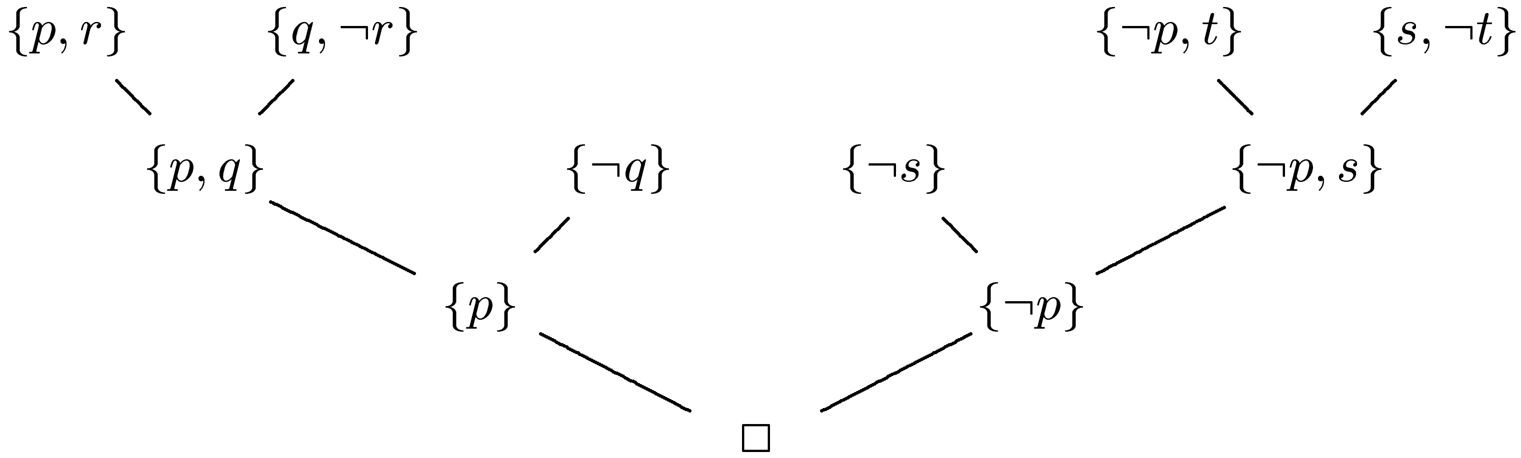
Rezoluce – stromy

- přehlednější formou rezolučních důkazů jsou stromy
- *strom rezolučního důkazu* C z S je binární strom T s vlastnostmi:
  - kořenem T je C
  - listy T jsou prvky S
  - libovolný vnitřní uzel C_i s bezprostředními následníky C_j, C_k je rezolventou C_j, C_k
- příklad: strom důkazu $C = \{\neg q\}$ z $S = \{\{p, r\}, \{\neg q, \neg r\}, \{\neg p\}\}$

$$
\begin{tikzcd}[arrows=dash]
\{p, r\} \arrow[dr] & & \{\neg q, \neg r\} \arrow[dl] & \\
& \{p, \neg q\} \arrow[dr] & & \{\neg p\} \arrow[dl] \\
& & \{\neg q\} &
\end{tikzcd}
$$

Rezoluce – příklad

- příklad: vytvořte strom rezolučního vyvrácení S (dokažte $S \vdash_R \square$), je-li $S = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{s, \neg t\}\}$



Rezoluce – vlastnosti

- *Věta (korektnost a úplnost rezoluce)*: rezoluční vyvrácení formule S existuje právě tehdy, když je S nesplnitelná.
- důsledek: existuje-li rezoluční strom s listy z množiny S a kořenem $\square$, pak je S nesplnitelná
- obecné schéma důkazu "formule A je log. důsledkem množiny formulí $\mathbf{T}$": vytvoříme konjunkci T' všech prvků z $\mathbf{T}$, formuli $T' \wedge \neg A$ převedeme do nkf a ukážeme nkf$(T' \wedge \neg A) \vdash_R \square$
- výhody pro strojové zpracování: systematické hledání důkazu, práce s jednoduchou datovou strukturou, jediné odvozovací pravidlo
- problém: strategie generování rezolvent – prohledávaný prostor může být příliš velký; př.: $\{\{p\}, \{p, \neg q\}, \{\neg p, q, r\}, \{\neg p, \neg r\}, \{r\}\}$ – postup, kdy rezolvujeme 2. a 3. klauzuli na p a výsledek poté se 4. na r, k důkazu nevede

Zjemnění rezoluce I

- snaha omezit prohledávaný prostor (přestože problém splnitelnosti formulí SAT je NP-úplný):
  - ukončením prohledávání neperspektivních cest
  - určením pořadí při procházení alternativních cest
- uvedená omezení vedou ke *zjemnění rezoluce*
- obecně mají omezení podobu dalších podmínek kladených na rodičovské klauzule nebo rezolventu v definici rezoluce
- každé omezení rezolučního pravidla je korektní (ale ne každé zachovává úplnost)

Zjemnění rezoluce II

- vyloučení klauzulí obsahujících literál, který se ve formuli S vyskytuje pouze v jedné paritě
- *T-rezoluce* jsou rezoluce, kde žádná z rodičovských klauzulí není tautologie
  - klauzule je tautologie, obsahuje-li týž literál v obou paritách (pozitivní bez negace i negativní s negací), např. $\{p, \neg q, \neg p, r\}$
  - T-rezoluce je korektní a úplná, tedy S je nesplnitelná $\Leftrightarrow S \vdash_{R^T} \square$
- nechť $\mathcal{A}$ je libovolná valuace, *$\mathcal{A}$-rezoluce (sémantická rezoluce)* je rezoluce, kde alespoň jedna z rodičovských klauzulí je v $\mathcal{A}$ nepravdivá
  - budou-li rodiče v dané valuaci pravdiví, bude v ní pravdivý i potomek – touto cestou k nesplnitelnosti nedojdeme
  - *$\mathcal{A}$-rezoluce* je korektní a úplná